

1 U.S. Patent Application Serial No. 09/599,806, entitled "Methods  
2 and Systems of Providing Information to Computer Users", bearing  
3 attorney docket number MS1-557us, and filed on the same date as  
4 this patent application;

- 5
- 6 • U.S. Patent Application Serial No. 09/599,299, entitled "Methods,  
7 Systems, Architectures and Data Structures For Delivering Software  
8 via a Network", bearing attorney docket number MS1-559us, and  
9 filed on the same date as this patent application;
  - 10 • U.S. Patent Application Serial No. 09/599,048, entitled "Network-  
11 based Software Extensions", bearing attorney docket number MS1-  
563us, and filed on the same date as this patent application;
  - U.S. Patent Application Serial No. 09/599,813, entitled "Authoring  
Arbitrary XML Documents Using DHTML and XSLT", bearing  
attorney docket number MS1-583us, and filed on the same date as  
this patent application;
  - U.S. Patent Application Serial No. 09/599,086, entitled "Task  
Sensitive Methods And Systems For Displaying Command Sets",  
bearing attorney docket number MS1-562us, and filed on the same  
date as this patent application.--

12

13

14 **In the Claims**

15 Claims 1, 9, 12, 18, 23, 30, 31 and 37 are amended.

16 Claims 1-42 are pending and are listed below as follows:

17

18 1. (Amended) A software architecture embodied on a computer-  
19 readable medium, the architecture comprising:

20 multiple attachment points collectively arranged to filter data associated  
21 with files that describe software extensions; and

22 multiple extension managers associated with the multiple attachment points  
23 and with respective feature types that can be added to a software platform by  
24 software extensions, the extension managers being configured to receive data from  
25

Sub B1  
A2

1 the multiple attachment points that pertains only to the feature type with which the  
2 extension manager is associated.

3  
4 2. The software architecture of claim 1, wherein the attachment points  
5 are defined as predicate chains.  
6

7 3. The software architecture of claim 1, wherein the attachment points  
8 filter XML data.  
9

10 4. The software architecture of claim 3, wherein each feature type is  
11 associated with an XML tag.  
12

13 5. The software architecture of claim 3, wherein each feature type is  
14 associated with an XML tag, at least some of the feature types comprising user-  
15 defined feature types.  
16

17 6. The software architecture of claim 1, wherein each attachment point  
18 exposes collections of ordered nodes.  
19

20 7. The software architecture of claim 1, wherein each attachment point  
21 exposes collections of ordered XML nodes.  
22

23 8. A computer embodying the software architecture of claim 1.  
24  
25

9. (Amended) A software architecture embodied on a computer-readable medium, the architecture comprising:

a hub structure configured to:

receive multiple different files that describe software extensions that can be added to a software platform;

combine the multiple different files into a single exposable list; and

expose the single exposable list to a filter structure that is configured to filter the list.

10. The software architecture of claim 9, wherein the hub structure receives multiple different XML files and exposes a list of XML nodes.

11. A computer embodying the software architecture of claim 9.

12. (Amended) A software architecture embodied on a computer-readable medium, the architecture comprising multiple different attachment points each of which is configured to:

receive XML data that pertains to one or more software extensions that can be added to a software platform;

process the XML data to provide a list of XML nodes; and

expose the list of XML nodes.

13. The software architecture of claim 12, wherein the list of XML nodes is exposed to another attachment point.

1           14.    The software architecture of claim 12, wherein the list of XML  
2 nodes can pertain to multiple different feature types that can be added by the one  
3 or more software extensions.  
4

5           15.    The software architecture of claim 12, wherein the list of XML  
6 nodes can pertain to multiple different features of particular feature types that can  
7 be added by the one or more software extensions.  
8

9           16.    The software architecture of claim 12, wherein the list of XML  
10 nodes can pertain to one or more of:

11               multiple different feature types that can be added by the one or more  
12 software extensions; and

13               multiple different features of particular feature types that can be added by  
14 the one or more software extensions.  
15

16           17.    A computer embodying the software architecture of claim 12.  
17

18           18.    (Amended) A software architecture embodied on a computer-  
19 readable medium, the architecture comprising:

20               a hub structure configured to:

21                   receive multiple different files that describe software extensions that  
22 can be added to a software platform;

23                   combine the multiple different files into a single exposable list; and

24                   expose the single exposable list to a filter structure that is configured  
25 to filter the list;

Sub  
B1  
a filter structure comprising multiple attachment points collectively  
arranged to filter data associated with the list exposed by the hub structure; and  
multiple extension managers associated with the multiple attachment points  
and with respective feature types that can be added to a software platform by  
software extensions, the extension managers being configured to receive data from  
the multiple attachment points that pertains only to the feature type with which the  
extension manager is associated.

19. The software architecture of claim 18, wherein the hub structure  
receives multiple different XML files and exposes a list of XML nodes.

20. The software architecture of claim 19, wherein the list contains root  
node tags for all of the XML files.

21. The software architecture of claim 19, wherein the XML files  
logically describe where a particular extension fits on the software platform.

22. The software architecture of claim 19, wherein the attachment points  
are defined as predicate chains.

Sub  
B1  
23. (Amended) The software architecture of claim 19, wherein an  
extension manager is notified whenever an extension comprising a feature type  
with which it is associated is added or removed from the software platform.

1           24.    The software architecture of claim 19, wherein each feature type is  
2 associated with a particular XML tag.

3  
4           25.    A computer embodying the software architecture of claim 18.

5  
6           26.    A method of providing a software extension comprising:  
7 exposing an XML list that contains one or more nodes;  
8 processing the XML list to identify specific nodes that correspond to  
9 various feature types that can be added to a software platform; and  
10 notifying an extension manager that is associated with at least one feature  
11 type if a node that corresponds to that feature type is identified in the XML list.

12  
13           27.    The method of claim 26, wherein said processing is accomplished by  
14 filtering the XML list using multiple attachment points that are defined as  
15 predicate chains.

16  
17           28.    The method of claim 27, wherein the individual attachment points  
18 receive XML data as an input and expose a list of XML nodes.

19  
20           29.    The method of claim 26, wherein said processing is accomplished by  
21 filtering on specific nodes.

22  
23           30.    (Amended) The method of claim 26, wherein said processing is  
24 accomplished by exposing various nodes.  
25

Sub  
B1  
AT

1 31. (Amended) The method of claim 26, wherein said processing is  
2 accomplished by filtering on specific nodes and exposing various nodes.

---

3  
4 32. One or more computer-readable media having computer-readable  
5 instructions thereon which, when executed by a computer, cause the computer to  
6 implement the method of claim 26.

7  
8 33. A method of providing a software extension comprising:  
9 receiving XML data that pertains to a software extension that is to be added  
10 to a software platform;  
11 processing the XML data to identify XML nodes; and  
12 exposing an XML list that contains one or more nodes that are identified by  
13 said processing.

14  
15 34. The method of claim 33, wherein said receiving comprises receiving  
16 multiple XML files that pertain to different software extensions.

17  
18 35. The method of claim 34, wherein said processing comprises  
19 combining the multiple XML files into a single exposable list.

20  
21 36. The method of claim 33, wherein said processing comprises  
22 processing the XML data with one or more attachment points that are defined as  
23 predicate chains that filter the XML data.

2 37. (Amended) The method of claim 36, wherein at least one of the  
attachment points exposes a node.

3  
4 38. The method of claim 36, wherein at least one of the attachment  
5 points filters on a node.

6  
7 39. One or more computer-readable media having computer-readable  
8 instructions thereon which, when executed by a computer, cause the computer to  
9 implement the method of claim 33.

10  
11 40. A method of providing a software extension comprising:  
12 receiving multiple different files, each of which being associated with a  
13 different software extension and logically describing its associated software  
14 extension;  
15 combining the multiple different files in a single list;  
16 exposing portions of the list;  
17 processing exposed portions of the list to identify one or more feature types  
18 that are to be added to a software platform; and  
19 notifying an extension manager that is associated with a particular feature  
20 type.

21  
22 41. The method of claim 40, wherein the multiple different files  
23 comprise XML files.



1           42. One or more computer-readable media having computer-readable  
2 instructions thereon which, when executed by a computer, cause the computer to  
3 implement the method of claim 40.  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25